

# 二分图覆盖和独立

2019年8月10日

17:29

◆

## ◆ 二分图覆盖

### 一. 二分图最小点覆盖集

1. 点覆盖集：包含图中所有边的至少一个端点的点集合
  - i. 最小点覆盖集：集合内点数最少的点覆盖集
2. König定理：二分图中， $|\text{最小点覆盖}| = |\text{最大匹配}|$ 
  - i. 匹配的关联点集无交集，因此 $|\text{最大匹配}| \leq |\text{最小点覆盖}|$
  - ii. 从最大匹配构造最小点覆盖的方法：从左部每个非匹配点出发，DFS求增广路，每次都更新DFS中访问过的所有左右部结点，则左部未标记点和右部标记点的并集即为最小点覆盖
    - 1) 这些点一定都是匹配点，否则就有增广路了，与最大匹配矛盾，因此两端非匹配点的边都会被该点集覆盖
    - 2) 每条匹配边的两端一定同时被访问或同时不被访问，因为DFS求增广路过程中，从右部点会到达左部匹配点，因此两端都是匹配点的边都会被该点集覆盖
    - 3) 左部非匹配点关联的匹配边所关联的右部点一定被访问而并在该点集中，因此左端非匹配点右端匹配点的边也被该点集覆盖
    - 4) 右部非匹配点关联的匹配边所关联的左部点一定未被访问而在该点集中，否则会有增广路，因此右端非匹配点左端匹配点的边也被该点集覆盖
    - 5) 综上， $|\text{最大匹配}| = |\text{最小点覆盖}|$
3. 二分图点覆盖模型
  - i. 2要素：每条边都有2个端点，至少选择其中1个

◆

## ◆ 二分图独立

### 二. 二分图最大点独立集

1. 点独立集：任两点在原图都不相邻的点集
  - i. 点独立集对应的导出子图上没有边，每个点都是孤立点
  - ii. 最大点独立集：集合内点数最多的点独立集
2. 团：任两点在原图都相邻的点集
  - i. 团对应的导出子图上全是边，是个完全图
  - ii. 最大团：集合内点数最多的团
3. Thm：无向图最大团 == 补图最大点独立集
  - i. 对一般无向图，最大团和最大独立集都是NPC问题
4. Thm：二分图中， $|\text{最大点独立集}| + |\text{最大匹配}| = \text{结点数}$ 
  - i. 原图的点集中，删除掉覆盖所有边的点之后，会使剩下的点都变成孤立点，因而， $|\text{最大点独立集}| + |\text{最小点覆盖集}| = \text{结点数}$ ，又因为上一节提过

的,  $|最大匹配| = |最小点覆盖|$ , 证毕

◆

◆ DAG覆盖

### 三. DAG的最小路径边覆盖集

#### 1. 拆点二分图

- i. 将DAG的 $n$ 个点拆成 $2n$ 个点, 左部点编号 $1 \sim n$ 对应右部点编号 $n+1 \sim 2n$
- ii. 对原图每条有向边 $(x,y)$ , 在左部点 $x$ 和右部点 $y+n$ 间连边
- iii. 相当于是让每条边的起点都放进左部, 终点放进右部
- iv. 得到的二分图称为原图 $G$ 的拆点二分图, 记作 $G_2$

#### 2. 最小路径覆盖: 在DAG上选择尽量少的**不相交**简单路径覆盖所有顶点

- i. Thm: 原图最小路径覆盖的边数  $=$  原图点数 $n$  -  $G_2$ 最大匹配数
- ii. 引: 不相交 iff 任意点入度和出度 $\leq 1$ ; 覆盖 iff 任意点入度+出度 $\geq 1$ ; 因而 $G$ 的最小路径覆盖子图对应 $G_2$ 的最大匹配子图, 即每条有向原边 $(x,y)$ 对应 $G_2$ 上的无向匹配边 $(x,y+n)$
- iii. 引:  $G$ 最小路径覆盖子图中, 0出度点 $x$  (每条简单路径的终点) 不会有有向边 $(x,y)$ 被选入覆盖, 也就对应上了 $G_2$ 中的非匹配点 $x$
- iv. 因而, 不相交简单路径条数最少 iff 终点最少 iff  $G_2$ 左部非匹配点最少

#### 3. 最小可重路径覆盖: 在DAG上选择尽量少的**可相交**路径覆盖所有顶点

- i. 即一个顶点可反复出现在一条路径上时的最小覆盖
- ii. 引: 形如 $upvxpy$ 这样覆盖两次 $p$ 的路径, 可通过让 $xpy$ 缩成 $xy$ 来避免重复
- iii. 不要求输出具体路径的话, 可以直接floyd传递闭包, 再求不可重覆盖, 不难发现这个做法的答案同时也是互不可达最大点集的大小
- iv. 例: [nanti.jisuanke.com/t/A1542](https://nanti.jisuanke.com/t/A1542) (南京2017现场赛M题): DAG最大互不可达点集大小